

BAB II

LANDASAN TEORI

2.1 Sistem Informasi Manajemen

2.1.1 Sistem

Beberapa pengertian atau definisi mengenai sistem yang diberikan oleh para ahli sebagai bahan perbandingan antara lain sebagai berikut:

- a. Sistem adalah satu kesatuan komponen yang saling terhubung dengan batasan yang jelas bekerja bersama-sama untuk mencapai seperangkat tujuan (O'Brien dan Marakas 2009).
- b. Pengertian sistem menurut Kadir (2003:54) sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai tujuan.
- c. Ackof dalam Effendy (1989:51) mengatakan bahwa sistem adalah setiap kesatuan, secara konseptual atau fisik, yang terdiri dari bagian-bagian dalam keadaan saling tergantung satu sama lain.
- d. Dalam Kamus Besar Bahasa Indonesia (1995:950) disebutkan bahwa sistem mempunyai dua pengertian; (a) Seperangkat unsur yang secara teratur saling berkaitan sehingga membentuk suatu totalitas; dan (b) Susunan yang teratur dari pandangan, teori, asas, dan sebagainya.

Dari definisi di atas, terlihat bahwa masing-masing menekankan bahwa sistem memakai pendekatan pada elemen atau komponen. Artinya, bahwa sistem haruslah terdiri atas berbagai komponen / elemen yang saling berhubungan sehingga membentuk satu kesatuan yang utuh.

2.1.2 Informasi

Informasi merupakan data atau fakta yang telah di proses sedemikian rupa, sehingga berubah bentuknya menjadi informasi. Di samping itu informasi dapat mengurangi ketidakpastian serta mempunyai nilai dalam keputusan karena dengan adanya informasi kita dapat memilih tindakan-tindakan dengan resiko yang paling kecil.

2.1.3 Manajemen

Setiap manusia dalam perjalanan hidupnya tidak akan terlepas dari yang namanya organisasi, dan di dalam organisasi itu dipelajarilah bagaimana cara mengelola organisasi dengan profesional. Ini menunjukkan ilmu manajemen bersifat universal. Oleh karena itu, salah satu cabang ilmu yang saat ini sangat berkembang pesat adalah ilmu manajemen. Ilmu ini telah melakukan intervensi ke berbagai bidang ilmu lain dalam pengembangannya. Kita kenal ada manajemen sumber daya manusia, manajemen perbankan, manajemen industri, manajemen keuangan, pemasaran, produksi, manajemen perkotaan, manajemen pemerintahan, manajemen pendidikan, dan sistem informasi manajemen. Secara umum pengertian manajemen adalah pengendalian dan pemanfaatan daripada semua factor dan sumber daya yang menurut suatu perencanaan (*planning*), diperlukan untuk mencapai atau menyelesaikan suatu tujuan-tujuan tertentu.

Raymond McLeod Jr (1996:54) mengemukakan bahwa Sistem Informasi Manajemen (SIM) adalah suatu sistem berbasis computer yang menyediakan informasi bagi beberapa pemakai dengan kebutuhan serupa. *Output* informasi digunakan oleh manajer maupun non manajer dalam perusahaan untuk membuat keputusan dalam memecahkan masalah [3].

2.2 Web Server

Web Server adalah komputer yang digunakan untuk menyimpan dokumen-dokumen web, komputer ini akan melayani permintaan dokumen web dari kliennya. *Browser web* seperti *Explorer* atau *Navigator* berkomunikasi melalui jaringan (termasuk jaringan internet) dengan *server web* menggunakan *HTTP*. Browser akan mengirimkan request kepada *server* untuk meminta dokumen tertentu atau layanan lain yang disediakan oleh *server*. *Server* memberikan dokumen atau layanannya jika tersedia juga dengan menggunakan protokol *HTTP*.

Fungsi utama dari *web server* yaitu untuk mentransfer atau memindahkan berkas yang diminta oleh pengguna melalui protokol komunikasi tertentu, karena dalam satu halaman *web* biasanya terdiri dari berbagai macam jenis berkas seperti

gambar, video, teks, audio, file dan lain sebagainya, maka pemanfaatan *web server* berfungsi juga untuk mentransfer keseluruhan aspek pemberkasan dalam halaman tersebut, termasuk teks, gambar, video, file dan sebagainya [4].

2.3 Bahasa Pemrograman *PHP*

PHP sendiri sebenarnya merupakan singkatan dari “*Hypertext Preprocessor*”, yang merupakan sebuah bahasa scripting tingkat tinggi yang dipasang pada dokumen HTML. Sebagian besar sintaks dalam *PHP* mirip dengan bahasa *C*, *Java* dan *Perl*, namun pada *PHP* ada beberapa fungsi yang lebih spesifik. Sedangkan tujuan utama dari penggunaan bahasa ini adalah untuk memungkinkan perancang web yang dinamis dan dapat bekerja secara otomatis. Pada awalnya *PHP* merupakan kependekan dari *Personal Home Page* (*Situs Personal*). *PHP* pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu *PHP* masih bernama *FI* (*Form Interpreted*), yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data form dari web.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya *PHP/FI*. Dengan perilis kode sumber ini menjadi *open source*, maka banyak *programmer* yang tertarik untuk ikut mengembangkan *PHP*. Pada November 1997, dirilis *PHP/FI 2.0*. Pada rilis ini interpreter *PHP* sudah menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk *PHP*: dan meresmikan rilis tersebut sebagai *PHP 3.0* dan singkatan *PHP* dirubah menjadi akronim berulang *PHP: Hypertext Preprocessing*. Pada pertengahan tahun 1999, Zend merilis interpreter *PHP* baru dan rilis tersebut dikenal dengan *PHP 4.0*. *PHP 4.0* adalah versi *PHP* yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi *web* kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi. Pada Juni 2004, Zend merilis *PHP 5.0*. Dalam versi ini, intinya dari interpreter *PHP* mengalami perubahan besar.

Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam *PHP* untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek [5].

2.4 Database MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data *SQL* (Bahasa Inggris: *Database Management System*) atau *DBMS* yang *multithread*, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. *MySQL AB* membuat *MySQL* tersedia sebagai perangkat lunak gratis dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan *GPL* (*General Public License*).

MySQL adalah *Relational Database Management System (RDBMS)* yang didistribusikan secara gratis dibawah lisensi *GPL*, dimana setiap orang bebas untuk menggunakan *MySQL*, namun tidak boleh dijadikan produk turunan yang bersifat komersial. *MySQL* sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu *SQL (Structured Query Language)*. *SQL* adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Keandalan suatu sistem *database (DBMS)* dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah *SQL*, yang dibuat oleh *user* maupun program-program aplikasinya. Sebagai *database server*, *MySQL* dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam query data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query MySQL* bisa sepuluh kali lebih cepat dari *PostgreSQL* dan lima kali lebih cepat dibandingkan *Interbase* [5].

2.5 Framework CodeIgniter

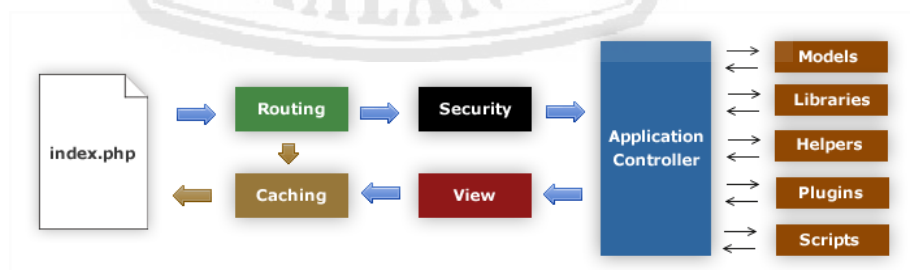
CodeIgniter adalah *framework* aplikasi web yang *open source* untuk bahasa pemrograman *PHP*. *CodeIgniter* memiliki banyak fitur yang membuatnya berbeda dengan *framework* lainnya. Tidak seperti beberapa *framework PHP* lainnya, dokumentasi untuk *framework* ini sangat lengkap, yang mencakup seluruh aspek dalam *framework*. *CodeIgniter* juga mampu berjalan pada

lingkungan *shared hosting* karena memiliki ukuran yang sangat kecil, namun memiliki kinerja yang sangat luar biasa.

Dari sisi pemrograman, *CodeIgniter* kompatibel dengan *PHP4* dan *PHP5*, sehingga akan berjalan dengan baik pada *web host* yang banyak dipakai pada saat ini. *CodeIgniter* menggunakan pola desain *Model-View-Controller (MVC)*, yang merupakan cara untuk mengatur aplikasi web kedalam tiga bagian yang berbeda, yaitu *Model* yaitu lapisan abstraksi *database*, *View* yaitu file-file template tampilan depan, dan *Controller* yaitu logika bisnis dari aplikasi. Pada intinya, *CodeIgniter* juga membuat penggunaan ekstensif dari pola desain *Singleton*. Maksudnya adalah cara untuk *me-load class* sehingga jika *class* itu dipanggil dalam beberapa kali, kejadian yang sama pada *class* tersebut akan digunakan kembali. Hal ini sangat berguna dalam koneksi *database*, karena kita hanya ingin menggunakan satu koneksi setiap kali *class* itu digunakan.

CodeIgniter dikembangkan oleh Rick Ellis, dengan versi awal yang dirilis pada tanggal 28 Februari 2006. Dari tahun itulah hingga sekarang, telah muncul banyak versi *CodeIgniter* yang terus berkembang dengan penambahan fitur baru dari versi sebelumnya [6].

Bagaimana suatu proses data mengalir pada sistem yang menggunakan *CodeIgniter Framework* dapat diilustrasikan pada gambar berikut:



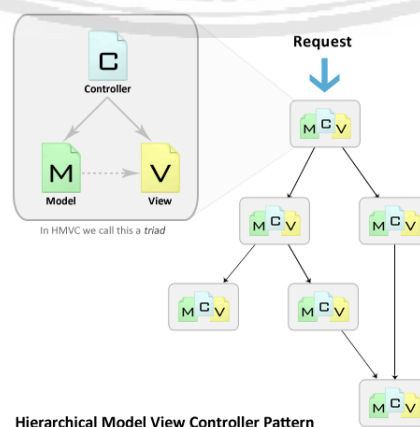
Gambar 2.1 Application Flow Diagram CodeIgniter Framework [7]

Keterangan:

1. **Index.php** berfungsi sebagai *front controller*, menginisialisasi *base resource* untuk menjalankan *CodeIgniter*.
2. **Router** memeriksa *HTTP Request* untuk menentukan apa yang harus dilakukan dengannya.
3. Jika **cache** aktif, maka hasilnya akan langsung dikirimkan ke *browser* dengan mengabaikan aliran data normal.
4. **Security**, sebelum *controller* dimuat, *HTTP request* dan data yang dikirimkan user akan difilter untuk keamanan.
5. **Controller** memuat *model*, *core libraries*, *plugins*, *helpers*, dan semua *resource* yang diperlukan untuk memproses *request*.
6. Akhirnya **View** yang dihasilkan akan dikirim ke *browser*. Jika *cache* aktif, maka *view* akan disimpan sebagai *cache* dahulu, sehingga pada *request* berikutnya langsung ditampilkan.

2.6 Konsep HMVC (Hierarchical Model-View-Controller)

HMVC merupakan konsep pengembangan design MVC (*model-view-controller*) yang diterapkan secara hirarkis. Keuntungan design ini adalah pada modularitas yang dikelompokkan berdasarkan fungsinya, atau lebih mudahnya dapat dikatakan setiap modul memiliki *file controller*, *model*, dan *view* masing-masing yang diletakkan dalam satu *folder*. Dengan konsep seperti ini memungkinkan pengembangan sistem semakin dalam [8].



Gambar 2.2 Diagram Konsep HMVC [8]

2.7 User Interface (UI)

Antarmuka pemakai (*User Interface*) merupakan mekanisme komunikasi antara pengguna (*user*) dengan sistem. Antarmuka pemakai dapat menerima informasi dari pengguna dan memberikan informasi kepada pengguna untuk membantu mengarahkan alur penelusuran masalah sampai ditemukan suatu solusi. *User Interface* berfungsi untuk menginputkan pengetahuan baru kedalam basis pengetahuan sistem pakar, menampilkan penjelasan sistem dan memberikan panduan pemakaian sistem secara menyeluruh *step by step* sehingga *user* mengerti apa yang akan dilakukan terhadap suatu sistem [9].

2.8 Metodologi Pengembangan Rekayasa Perangkat Lunak

2.8.1 XP (*Extreme Programming*)

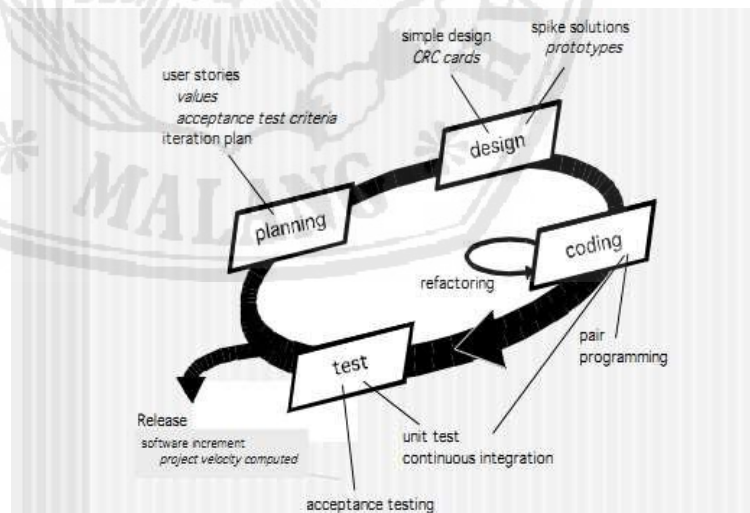
Extreme Programming merupakan salah satu metodologi dalam pengembangan rekayasa perangkat lunak dan juga merupakan salah satu dari beberapa *agile software development methodologies* yang berfokus pada *coding* sebagai aktivitas utama disemua tahap pada siklus pengembangan yang lebih responsive terhadap kebutuhan *customer* (*'agile'*) dibandingkan dengan metode-metode tradisional sambil membangun suatu *software* dengan kualitas yang lebih baik, selain itu *Extreme Programming* meliputi seluruh area pengembangan perangkat lunak. Model *agile* proses ini dikembangkan oleh Kent Beck dan Ward Cunningham pada bulan Maret 1996. Model *extreme programming* ini merupakan yang terpopuler dari beberapa metodologi pengembangan *software* yang dipakai untuk mengimplementasi proyek pengembangan perangkat lunak.

Extreme programming menggunakan pendekatan berorientasi objek sebagai paradigma pembangunan yang lebih disukai dan meliputi satu *set rules* dan *practices* yang berlaku dalam konteks dari 4 aktifitas *framework*:

- a. **Planning**, aktifitas yang juga disebut sebagai *the planning game* dimulai dengan “mendengarkan” yaitu sebuah aktifitas mengumpulkan kebutuhan yang memungkinkan para anggota teknikal dari tim *XP* untuk memahami konteks bisnis untuk perangkat lunak dan untuk mendapatkan *broad feel* untuk *output*

yang dibutuhkan dan fitur utama serta fungsionalitas. Pada tahap ini, *stakeholder* dan *programmer* bekerjasama untuk menentukan bagaimana mengelompokkan cerita ke dalam rilis berikutnya atau peningkatan perangkat lunak selanjutnya, yang akan dibangun oleh tim *XP*.

- b. **Design**, menyediakan panduan implementasi untuk proses seperti yang sudah dituliskan. Gagasan pusat di dalam *XP* adalah *design* terjadi sebelum dan setelah *coding* dimulai.
- c. **Coding**, menterjemahkan penjabaran yang sudah dilakukan pada tahap penulisan kode program.
- d. **Testing**, pada tahap ini *unit test* yang dikreasikan harus diimplementasikan menggunakan *framework* yang memungkinkan mereka menjadi otomatis (karenanya, dapat dieksekusi dengan mudah dan berulang-ulang). Hal ini mendorong strategi regresi testing ketika kode dimodifikasi [9].



Gambar 2.3 Extreme Programming [9]

2.8.2 Prinsip Dasar *Extreme Programming*

Terdapat lima prinsip dasar yang sangat fundamental dalam *Extreme Programming*, dimana prinsip-prinsip ini digunakan untuk menentukan apakah semua tindakan / pekerjaan yang telah dilakukan akan selalu sukses atau sebaliknya (dalam konteks *Extreme Programming*). Kelima prinsip tersebut adalah:

1. Aliran umpan balik (*Rapid Feedback*)
2. Asumsi kesederhanaan (*Assume Simplicity*)
3. Penambahan perubahan (*Incremental Change*)
4. Pemeluk pekerjaan (*Embrace Work*)
5. Kualitas kerja (*Quality Work*)

2.8.3 Kunci Utama XP

XP sebagai sebuah metode yang dinamis diperlihatkan dalam empat values yang dimilikinya dan keempatnya merupakan dasar-dasar yang diperlukan dalam XP, Kent Beck menyatakan bahwa tujuan jangka pendek individu sering berbenturan dengan tujuan sosial jangka panjang. Karena itu dibuatlah *values* yang menjadi aturan, hukuman, dan juga penghargaan. Keempat *values* tersebut adalah:

1. Komunikasi (*Communication*)

Komunikasi menekankan pada pendekatan yang lebih menekankan pada orang per orang secara langsung, dimana hal tersebut akan lebih baik daripada hanya berdasarkan pada dokumen yang menjelaskan tentang *software* yang dibangun. Dan juga, kemungkinan penggunaan dari beberapa praktis XP sehingga dibutuhkan alokasi waktu yang banyak untuk berkomunikasi dengan *customer*.

2. Kesederhanaan (*Simplicity*)

Simplicity adalah sebuah nilai (*value*) dari XP yang digunakan untuk memberikan solusi dari *problem* atau permasalahan yang

dihadapi oleh customer sehingga problem atau masalah tersebut bisa disederhanakan. Kedua pihak yaitu *developer* dan *customer* bisa mengerti solusi *software* jika itu tidak nyata, dimana solusi tersebut merupakan kandidat untuk praktis yang lain yang disebut dengan *refactoring* (salah satu inti dari praktis XP).

3. Umpan Balik (*Feedback*)

Feedback berarti bahwa segala sesuatu yang telah dilakukan atau dicapai dievaluasi dengan respek / reaksi untuk mengetahui bagaimana agar pekerjaan tersebut berjalan dengan baik dan menghasilkan *software* yang sesuai dengan kebutuhan *customer*. Pernyataan bagaimana agar pekerjaan tersebut berjalan dengan baik mengindikasikan bahwa *feedback* diperoleh dari evaluasi setiap bagian pekerjaan dari solusi. *Feedback*, yang merupakan hasil kerja dari Nyquist di Laboratorium Bell pada awal tahun 1930, digunakan untuk menjamin bahwa solusi tersebut adalah benar. Pada saat *engineers* (insinyur) mulai mempelajari *feedback*, hal tersebut telah menjadi salah satu jalan didalam mencari solusi dari suatu masalah.

4. Keberanian (*Courage*)

Courage berarti bahwa pihak pengembang mempersiapkan segala sesuatunya untuk membuat keputusan yang sangat penting yang mendukung praktis *XP* pada saat membangun dan merilis (meluncurkan) *software* kepada *customer* untuk masing-masing iterasi [9].

2.8.4 Kelebihan *XP*

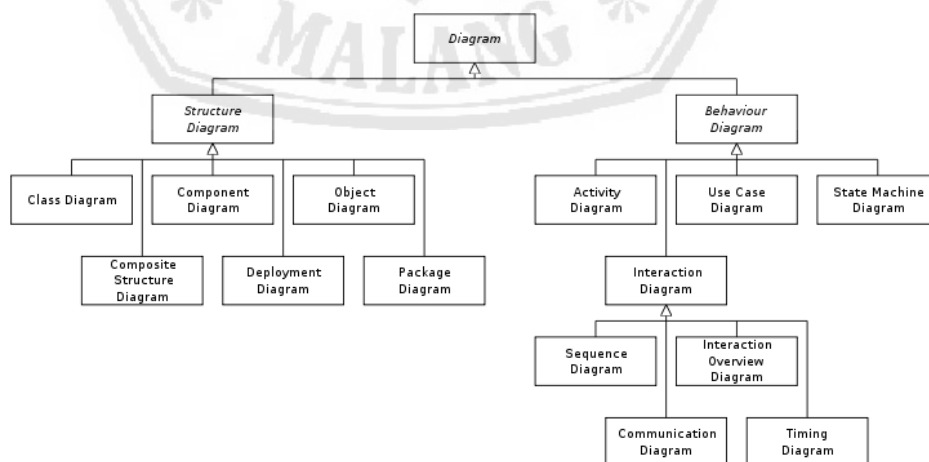
Adapun kelebihan dari pemakaian *Extreme Programming* dibandingkan dengan model pengembangan perangkat lunak yang lain adalah sebagai berikut:

Extreme Programming adalah metode yang semi-formal yang artinya metode ini bersifat fleksibel dan perubahan akan selalu diterima oleh developer tidak seperti pada metode *Waterfall* yang tidak fleksibel sehingga akan sulit mengadaptasinya ketika terjadi perubahan [9].

2.9 Unified Modeling Language (UML)

UML adalah sebuah standarisasi pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek. *UML* muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak. *UML* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

Model *UML* terdiri atas banyak elemen-elemen grafis yang digabungkan membentuk diagram. Tujuan representasi elemen-elemen grafis ke dalam diagram adalah untuk menyajikan beragam sudut pandang dari sebuah sistem berdasarkan fungsi masing-masing diagram tersebut. Kumpulan dari beragam sudut pandang inilah yang disebut sebuah model. *UML* mendefinisikan diagram-diagram sebagai berikut:



Gambar 2.4 Structure UML [10]

2.10 Aturan Pengelompokan Otomatis

Pengelompokan otomatis merupakan salah satu fitur utama pada sistem ini, agar sistem dapat melakukan pengelompokan peserta secara otomatis, maka dibutuhkan aturan membagi rata data mahasiswa agar didapatkan hasil pengelompokan peserta yang seimbang dan merata.

Berikut ini merupakan alur / tata cara bagaimana membagi rata peserta kkn. Terdapat dua atribut dari mahasiswa yang akan dijadikan variabel, yaitu jurusan dan jenis kelamin. Terdapat dua tahapan pembagian, yaitu pembagian jenis kelamin, dan pembagian jurusan. Berikut merupakan contoh implementasi aturan pengelompokan otomatis disertai dengan contoh kasus.

Sebagai contoh, pada suatu periode terdapat data mahasiswa pendaftar kkn sejumlah 1175 mahasiswa, dengan rincian jurusan dan jenis kelamin sebagai berikut.

Tabel 2.1 Contoh Data Mahasiswa Pendaftar KKN

Jurusan	Jenis Kelamin		Jumlah
	L	P	
Teknik Informatika	105	80	185
Ilmu Komunikasi	82	113	195
Bahasa Indonesia	67	106	173
Ilmu Hukum	75	47	122
Akuntansi	97	63	160
Teknik Mesin	125	50	175
Peternakan	122	43	165
TOTAL	673	502	1175

Kemudian pada periode tersebut terdapat 40 lokasi kkn yang tersedia, selanjutnya data mahasiswa tersebut akan dibagi rata ke seluruh kelompok dengan menggunakan perhitungan berikut.

2.10.1 Pembagian Jumlah Kelompok

Jumlah kelompok kkn yang akan dibentuk bergantung pada jumlah desa / lokasi yang tersedia pada suatu periode, tersedia dalam artian telah mendapat izin dari kepala daerah / kepala desa setempat untuk dilaksanakannya kegiatan kkn. Dalam contoh kasus ini terdapat 40 desa yang tersedia, artinya terdapat 40 kelompok yang akan dibentuk, yang mana sebuah kelompok akan menempati satu desa, maka dapat dituliskan sebagai rumus berikut.

$$JK = JL$$

..... **Rumus 2.1**

Keterangan:

JK = Jumlah Kelompok

JL = Jumlah Lokasi

2.10.2 Pembagian Berdasarkan Jenis Kelamin

Mengacu pada data pendaftar kkn yang tertera pada **Tabel 2.1**, total jumlah mahasiswa laki-laki adalah sebanyak 673 dan perempuan sebanyak 502. Untuk mendapatkan sebuah kelompok dengan pembagian jenis kelamin yang ideal, perlu dilakukan perhitungan yaitu dengan menggunakan rumus, hasil perhitungan akan dijabarkan sebagai berikut.

a. Rumus Pembagian Mahasiswa Laki-laki

$$AngL = MhsL \text{ DIV } JK$$

..... **Rumus 2.2**

Keterangan:

AngL = Anggota Kelompok Laki-laki

MhsL = Mahasiswa Pendaftar Laki-laki

JK = Jumlah Kelompok

Hasil perhitungan:

$$\begin{aligned} AngL &= 673 \text{ DIV } 40 \\ &= 16 \text{ (16 orang)} \end{aligned}$$

Dari hasil perhitungan diatas, dapat disimpulkan bahwa sebuah kelompok yang ideal adalah dengan jumlah laki-laki sebanyak kurang lebih 16 orang.

b. Rumus Pembagian Mahasiswa Perempuan

$$\text{AngP} = \text{MhsP} \text{ DIV } \text{JK}$$

..... **Rumus 2.3**

Keterangan:

AngP = Anggota Kelompok Perempuan

MhsP = Mahasiswa Pendaftar Perempuan

JK = Jumlah Kelompok

Hasil perhitungan:

$$\begin{aligned}\text{AngP} &= 502 \text{ DIV } 40 \\ &= 12 \text{ (12 orang)}\end{aligned}$$

Dari hasil perhitungan diatas, dapat disimpulkan bahwa sebuah kelompok yang ideal adalah dengan jumlah perempuan sebanyak kurang lebih 12 orang. Untuk menentukan jumlah anggota yang ideal untuk sebuah kelompok maka hasil dari kedua perhitungan tersebut dijumlahkan sebagai berikut:

$$\begin{aligned}\text{AngL} + \text{AngP} &= 16 + 12 \\ &= 28 \text{ (28 orang)}\end{aligned}$$

Dari hasil perhitungan diatas dapat disimpulkan bahwa syarat sebuah kelompok agar dapat dikatakan ideal adalah yang memiliki jumlah anggota 28 orang, yang terdiri dari 16 orang laki-laki dan 12 orang perempuan.

2.10.3 Pembagian Berdasarkan Jurusan

Perhitungan ini digunakan untuk mendapatkan nilai ideal pembagian mahasiswa dalam sebuah kelompok berdasarkan jurusannya. Mengacu pada **Tabel 2.1**, pada contoh kasus ini dapat diketahui bahwa dari seluruh data pendaftar kkn tersebut berasal dari 7 ragam jurusan, yaitu teknik informatika, ilmu komunikasi, bahasa indonesia, ilmu hukum, akuntansi, teknik mesin, dan peternakan. Hasil perhitungan akan dijabarkan sebagai berikut.

$$\text{AngJur} = \text{MhsJurL} \text{ DIV } \text{JK}$$

..... **Rumus 2.4**

Keterangan:

AngJurL = Anggota laki-laki dari sebuah jurusan

MhsJurL = Pendaftar laki-laki dari sebuah jurusan

DIV = Hasil Bagi

JK = Jumlah Kelompok

$$\text{AngJurP} = \text{MhsJurP} \text{ DIV } \text{JK}$$

..... **Rumus 2.5**

Keterangan:

AngJurP = Anggota perempuan dari sebuah jurusan

MhsJurP = Pendaftar perempuan dari sebuah jurusan

DIV = Hasil Bagi

JK = Jumlah Kelompok

a. Pembagian Jurusan Teknik Informatika

Hasil perhitungan laki-laki:

$$\text{AInfL} = 105 \text{ DIV } 40$$

$$= 2 \text{ (2 orang)}$$

Hasil perhitungan perempuan:

$$\text{AInfP} = 80 \text{ DIV } 40$$

$$= 2 \text{ (2 orang)}$$

Keterangan:

AInfL = Anggota Informatika Laki-laki

AInfP = Anggota Informatika Perempuan

Kesimpulan:

Setiap kelompok yang ideal akan memiliki anggota dari jurusan informatika dengan jumlah kurang lebih 2 orang laki-laki dan 2 orang perempuan.

b. Pembagian Jurusan Ilmu Komunikasi

Hasil perhitungan laki-laki:

$$\begin{aligned} \text{AKomL} &= 82 \text{ DIV } 40 \\ &= 2 \text{ (2 orang)} \end{aligned}$$

Hasil perhitungan perempuan:

$$\begin{aligned} \text{AKomP} &= 113 \text{ DIV } 40 \\ &= 2 \text{ (2 orang)} \end{aligned}$$

Keterangan:

AKomL = Anggota Komunikasi Laki-laki

AKomP = Anggota Komunikasi Perempuan

Kesimpulan:

Setiap kelompok yang ideal akan memiliki anggota dari jurusan komunikasi dengan jumlah kurang lebih 2 orang laki-laki dan 2 orang perempuan.

c. Pembagian Jurusan Bahasa Indonesia

Hasil perhitungan laki-laki:

$$\begin{aligned} \text{ABinL} &= 67 \text{ DIV } 40 \\ &= 1 \text{ (1 orang)} \end{aligned}$$

Hasil perhitungan perempuan:

$$\begin{aligned} \text{ABinP} &= 106 \text{ DIV } 40 \\ &= 2 \text{ (2 orang)} \end{aligned}$$

Keterangan:

ABinL = Anggota Bahasa Indonesia Laki-laki

ABinP = Anggota Bahasa Indonesia Perempuan

Kesimpulan:

Setiap kelompok yang ideal akan memiliki anggota dari jurusan bahasa indonesia dengan jumlah kurang lebih 1 orang laki-laki dan 2 orang perempuan.

d. Pembagian Jurusan Ilmu Hukum

Hasil perhitungan laki-laki:

$$AHukL = 75 \text{ DIV } 40$$

$$= 1 \text{ (1 orang)}$$

Hasil perhitungan perempuan:

$$AHukP = 47 \text{ DIV } 40$$

$$= 1 \text{ (1 orang)}$$

Keterangan:

AHukL = Anggota Hukum Laki-laki

AHukP = Anggota Hukum Perempuan

Kesimpulan:

Setiap kelompok yang ideal akan memiliki anggota dari jurusan ilmu hukum dengan jumlah kurang lebih 1 orang laki-laki dan 1 orang perempuan.

e. Pembagian Jurusan Akuntansi

Hasil perhitungan laki-laki:

$$AAknL = 97 \text{ DIV } 40$$

$$= 2 \text{ (2 orang)}$$

Hasil perhitungan perempuan:

$$AAknP = 63 \text{ DIV } 40$$

$$= 1 \text{ (1 orang)}$$

Keterangan:

AAknL = Anggota Akuntansi Laki-laki

AAknP = Anggota Akuntansi Perempuan

Kesimpulan:

Setiap kelompok yang ideal akan memiliki anggota dari jurusan akuntansi dengan jumlah kurang lebih 2 orang laki-laki dan 1 orang perempuan.

f. Pembagian Jurusan Teknik Mesin

Hasil perhitungan laki-laki:

$$\begin{aligned} \text{AMsnL} &= 125 \text{ DIV } 40 \\ &= 3 \text{ (3 orang)} \end{aligned}$$

Hasil perhitungan perempuan:

$$\begin{aligned} \text{AMsnP} &= 50 \text{ DIV } 40 \\ &= 1 \text{ (1 orang)} \end{aligned}$$

Keterangan:

AMsnL = Anggota Mesin Laki-laki

AMsnP = Anggota Mesin Perempuan

Kesimpulan:

Setiap kelompok yang ideal akan memiliki anggota dari jurusan mesin dengan jumlah kurang lebih 3 orang laki-laki dan 1 orang perempuan.

g. Pembagian Jurusan Peternakan

Hasil perhitungan laki-laki:

$$\begin{aligned} \text{APtnL} &= 122 \text{ DIV } 40 \\ &= 3 \text{ (3 orang)} \end{aligned}$$

Hasil perhitungan perempuan:

$$\begin{aligned} \text{APtnP} &= 43 \text{ DIV } 40 \\ &= 1 \text{ (1 orang)} \end{aligned}$$

Keterangan:

APtnL = Anggota Peternakan Laki-laki

APtnP = Anggota Peternakan Perempuan

Kesimpulan:

Setiap kelompok yang ideal akan memiliki anggota dari jurusan peternakan dengan jumlah kurang lebih 3 orang laki-laki dan 1 orang perempuan.

Untuk mengetahui apakah hasil perhitungan yang telah dilakukan sesuai, maka perlu dilakukan penjumlahan seluruh variabel hasil perhitungan, hasilnya adalah seperti berikut:

$$\begin{aligned}\text{Kelompok ideal} = & (A_{\text{InfL}} + A_{\text{InfP}}) + (A_{\text{KomL}} + A_{\text{KomP}}) \\ & + (A_{\text{BinL}} + A_{\text{BinP}}) + (A_{\text{HukL}} + A_{\text{HukP}}) \\ & + (A_{\text{AknL}} + A_{\text{AknP}}) + (A_{\text{MsnL}} + A_{\text{MsnP}}) \\ & + (A_{\text{PtnL}} + A_{\text{PtnP}})\end{aligned}$$

$$\begin{aligned}= & (2 + 2) + (2 + 2) \\ & + (1 + 2) + (1 + 1) \\ & + (2 + 1) + (3 + 1) \\ & + (3 + 1)\end{aligned}$$

$$= 24$$

Dari hasil perhitungan diatas didapatkan nilai akhir sebesar 24, artinya sebuah kelompok dapat dikatakan ideal jika memiliki anggota kelompok berjumlah kurang lebih 24 orang yang terdiri dari berbagai jurusan (dalam kasus ini 7 ragam jurusan). Demikian perhitungan dengan model ini akan terus dilakukan hingga iterasi terakhir, yaitu iterasi ke - 40 (kelompok ke - 40).